Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 50 (2015) 363 – 368

Procedia Computer Science

2nd International Symposium on Big Data and Cloud Computing (ISBCC'15)

Variant of COBWEB Clustering for Privacy Preservation in Cloud DB Querying

Nazneen Mulani[a], Ambika Pawar[b], Preeti Mulay[c], Ajay Dani[d,]*

[a]*M .Tech Student, Symbiosis Institute of Technology, Pune, Gram-Lavale, Tal-Mulshi, 412115, India*
[b]*Researcher Scholar, Symbiosis Institute of Technology, Pune, Gram-Lavale, Tal-Mulshi, 412115, India*
[c]*Research Guide, Symbiosis Institute of Technology, Pune, Gram-Lavale, Tal-Mulshi, 412115, India*
[d]*Research Guide, Symbiosis Symbiosis Institute of Technology, Pune, Gram-Lavale, Tal-Mulshi, 412115, India*

Abstract

Cloud computing security is the set of control-based technologies and policies designed to protect information, data and infrastructure mapped with cloud computing applications. As Cloud computing includes shared resource, identity management, privacy and access control over network, it is necessary to provide privacy in these and other potentially vulnerable areas. This paper concentrate on achieving Privacy Preservation in Cloud DB using Incremental conceptual Clustering algorithm: Variant of COBWEB in replacement of incremental k-means. Variant of COBWEB is employed at Cloud Database Owner to secure Clients data at backend. Sample CRM dataset from UCI repository is analyzed in WEKA for comparative study of both algorithms.

Keywords: incremental conceptual Clustering, Variant of COBWEB, incremental k-means, UCI repository .

1. Introduction

Cloud environment provides access to dynamically scalable and virtualized resources to the user over the Internet. Database security is a challenge due to virtual set up and use over the internet. Many SaaS companies offer CRM services through their multitenant shared facilities so clients can manage their customers without buying software. These represent only the beginning of options for delivering all kinds of complex capabilities to both businesses and individuals. In SaaS based environment, Cloud Security plays major role. Security is critical due to the varied

* Nazneen Mulani. Tel.: +91 8007824322.
E-mail address: nazneen.mulani@sitpune.edu.in

Peer-review under responsibility of scientific committee of 2nd International Symposium on Big Data and Cloud Computing (ISBCC'15)
doi:10.1016/j.procs.2015.04.034

services that can be provided through a cloud. Privacy preservation model of cloud [6] uses the Encryption and Decryption algorithm on Database tables to protect data from unauthorized access and many unwanted security attacks. In real time, when any user fires a query to access data from Cloud database [6], it works as follows as shown in Fig. 1. Database Owner encrypts database records and sends it to Server using Encrypt algorithm, when Server is offline. While Database Owner runs Set Up algorithm to initialize some parameters and decryption key by running Extract algorithm with Database Owner. When any user fires the query, it obtains search token and decryption key from Database Owner. Then user sends token to Server who uses the token to search on each encrypted database records, for which Server runs Test Algorithm. In Paper [1], solution for enhancing privacy preservation in the existing model with k-means clustering is presented to find vulnerable attributes.

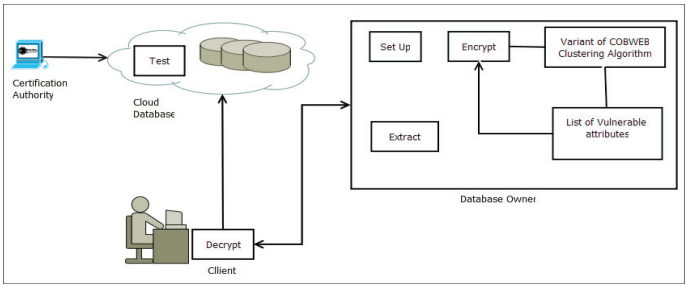

Fig. 1. Cloud Privacy Preservation Model

2. Variant of COBWEB Clustering Algorithm

The COBWEB algorithm was published by Fisher [4] in (1987). The key component of the COBWEB algorithm is the measure of similarity which is used to establish relationships between instances. Both the addition function and the mechanism used to search for instances within the tree, employ a heuristic measure called the category utility. Category utility is a measure of similarity between instances, and therefore acts as a measure of the quality of a given cluster. The category utility is represented by the result of a calculation which takes account of each attribute in an instance, comparing it to the attribute values of the other instances within a category. In this proposed system**, Variant of COBWEB algorithm** is presented and employed in existing Privacy Preservation scheme.

2.1. COBWEB with Category Utility Function

Category utility [4] was described by Gluck and Corter (1985) as a method for the creation of basic categories in a similar manner to those created by the human brain. A basic level category is defined as one which is preferred to a more generalized or specific category during object recognition. For example, “employee” in preference to other categories’ labels such as “Department” (more general) or “Company” (more specific). The formula for Category Utility is defined in Equation 1.

$$CU = \frac{1}{n}\sum_{k}^{n} P(C_k)\sum_{i}\sum_{j}\left[P\left(A_i=V_{ij}|C_k\right)^2 - P\left(A_i=V_{ij}\right)^2 \right] \quad (1)$$

Where, $A_i = V_{ij}$ represents Attribute value and C_k represents Classes. Intra-cluster Similarity is defined by $P\left(A_i=V_{ij}|C_k\right)$, Larger this probability the greater the proportion of class members sharing the value $\left(V_{ij}\right)$and more predictable the value of class members. $P\left(A_i=V_{ij}\right)$ is the probability of feature A_i taking on value V_{ij} . Here n indicates number categories. The category utility measure can be most easily understood as a type of distance measure. The output from the calculation determines whether or not an instance is enough ‘a-like’ the other

instances within a cluster to be made a member of that category itself, while creating dendogram. Variant of COBWEB is applied in Privacy preservation Model in replacement of k-means algorithm [1], to find the vulnerable attributes in existing database on cloud. In proposed system, the attributes which are repeating more are treated as Vulnerable attributes. As COBWEB algorithm constructs a classification tree incrementally by inserting the objects into classification tree one by one. After inserting the objects into classification tree COBWEB traverses the tree for balancing. COBWEB operates in bidirectional using different operations like 'merge', 'split' etc. In CU formula, number of occurrences of specific attributes is indicated by $P\left(A_i = V_{ij} \mid C_k\right)$ and $P\left(A_i=V_{ij}\right)$. As the number of occurrences of specific attribute value increases, CU value also increases. Indirectly, this specific indicates the vulnerability of attribute due to repetition. In next phase, all these vulnerable attributes are encrypted at Data Base Owner level Consider the Following Table 1. as example, which categorize the data using CU function calculation.

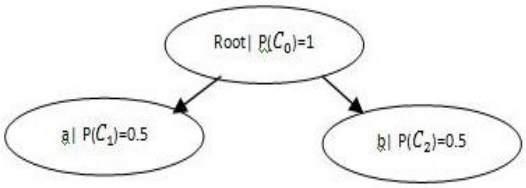

Fig. 2. Classification Tree

Table. 1. Vehicle Dataset

	color	Wheels	Passenger Capacity
a	White	2	1
b	White	2	2
c	Black	2	2
d	Black	3	1

As shown in Table. 1., Dataset contains total 4 instances with 3 attributes. As COBWEB constructs a Classification tree based on CU values as in Fig. 2., first it starts with Root Node and operates in incremental manner to construct whole tree with balanced CU factor. After incorporating instance a and b, tree structure looks like follows and Table 2 indicates representation of root node and its content after first iteration.

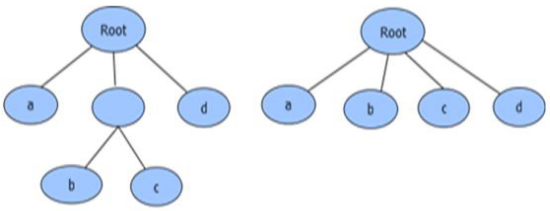

Fig.. 3. Spurious Node removal representation

Table. 2. CU calculation for Root Node

C_k		$P(C_0)=1$
Attribute	value	P
color	White	1.0
	Black	0.0
wheels	1	0.0
	2	1.0
	3	0.0
Passenger capacity	1	0.5
	2	0.5

2.2. Modified Category Utility Function

In this Category Utility of COBWEB is modified [3] in Variant of COBWEB to overcome the disadvantages of existing COBWEB [4]. Addition of terms in CU function implies to increase intra-cluster similarity and also

prevents formation of skew spurious intermediate nodes without a loss of learning accuracy as shown in Fig. 3.. It makes classification tree more balanced and also reduces number of uses of bidirectional operators as the spurious nodes decreases in the tree. Where N_j represents the number of nominal attributes A_i can have in given dataset.

$$CU = \frac{1}{n}\sum_{k}^{n} P(Ck)\sum_{i}\sum_{j}\left[\left(P(Ai = Vij \mid Ck) - \frac{1}{Nj}\right)\left\{P(Ai = Vij \mid Ck)^2 - P(Ai = Vij)^2\right\}\right] \quad (2)$$

2.3. Principal Component Analysis on COBWEB

When PCA applied on dataset it tries to identify the components that characterize the data. It is a multivariate technique that analyzes a data table in which observations are described by several inter-correlated quantitative dependent variables. The goals of PCA is extract the most important information from the data table[7] . PCA is applied on dataset using WEKA to find the impactful attributes from database stored at Database Owner. Mathematically, PCA is defined as a orthogonal linear transformation and assumes all basis vectors are an orthogonal matrix and concerned with finding the variances and coefficients of a dataset by finding the correlation matrix.

2.4. Mapping with Privacy Preservation Model

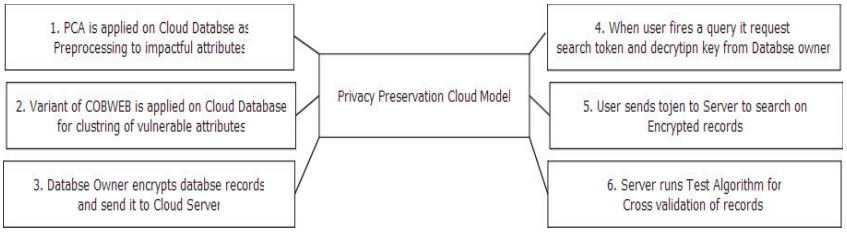

Fig. 4. Basic Steps in Privacy Preservation Model

Above Fig. 4 describes step wise solution of privacy preservation model. When any user fires a query on Cloud Database, privacy preservation models runs through various stages. Variant of COBWEB is applied to find vulnerability of attributes in Cloud Database. Preprocessing is carried out on Database, before applying Variant of COBWEB with modified CU function. Following steps describes all mentioned steps in brief.

1. Principal Component Analysis is applied as preprocessing to find characterized attribute from Cloud Database Table, which reduces the memory load of carrying extra attributes.
2. Variant of COBWEB is applied on extracted impactful attributes, to find vulnerability of attributes from Cloud Database.
3. When Server is offline, Database Owner encrypts database records using Encrypt algorithm and sends it to Server. When user fires a query, encrypted records is made available.
4. While Database Owner runs Set Up algorithm to initialize some parameters and decryption key by running Extract algorithm with Database Owner.
5. When any user fires the query, it obtains search token and decryption key from Database Owner and same key is used for future use.
6. Then user sends token to Server who uses the token to search on each encrypted database records, for which Server runs Test Algorithm.

4. Comparison of Variant of COBWEB and incremental k-means in WEKA Tool

'Wholesale Customer Data Set' from UCI machine learning repository [8] is used with data mining tool WEKA 3.7. Wholesale customer dataset is about the items purchased by Customer in wholesale retail Price with channel and region. This data set contains 8 attribute with 440 instances, all 8 are considered as valid attributes for further clustering and further PCA is applied with Variant of COBWEB to get impactful attributes.

Table. 3. Wholesale Customer Data Set Analysis in WEKA

Iteration	Total instances	Incremental k-means/time in sec	Variant of COBWEB/time in sec
1	249	14/0.03	16/0.02
2	249+151	19/0.06	19/0.03
3	249+151+40	21/0.07	23/0.04

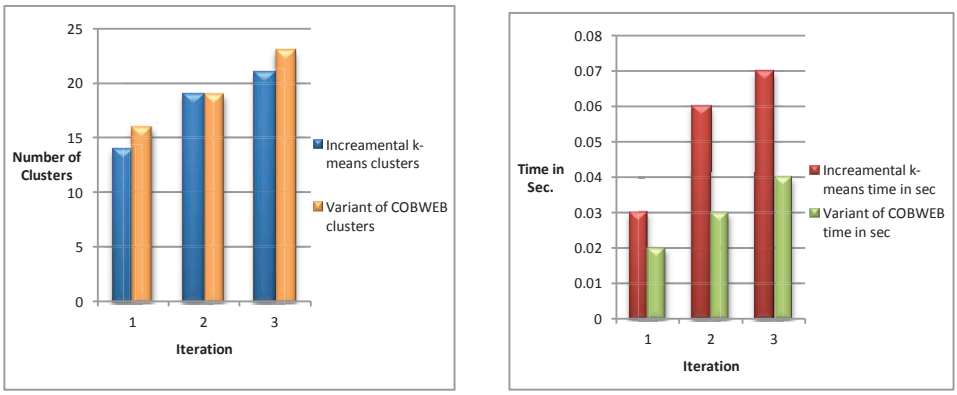

Fig. 5. Comparative Analysis in WEKA Tool

For the given data set the incremental k-means and COBWEB is analyzed based on the number of clusters produced by each individual algorithm along with the time needed to form the clusters. As COBWEB is independent of 'k' i.e. no of clusters, it works in incremental manner and generates hierarchical clustering [2]. In case of incremental k-means clustering, first hierarchical clustering is used in WEKA on same dataset to obtain the value of 'k' and in next phase same value is fed for k-means clustering which is renamed as incremental k-means. As hierarchical clustering is classic agglomerative (i.e. bottom up) hierarchical clustering method which forms the dendogram in first iteration for the input data. Based on the given input hierarchical clustering forms the clusters in dendogram form, that number of clusters will be considered as input to k-means. This technique is considered as validation of k-means clustering [5] in cluster formation technique. Table. 3. Shows the results obtained for different number of instances in particular iteration. As time needed for incremental k-means clustering is more than the time needed for COBWEB clustering. Also cluster formation of COBWEB mechanism is good and produces best clustering as compared to k-means.

5. Conclusions

This paper solves the problem of Privacy preservation in Cloud database querying using Variant of COBWEB clustering algorithm. Identification of Vulnerable attributes from Cloud database provides security for Clients data by preventing Brute force attacks. Comparative analysis of incremental k-means and Variant of COBWEB in WEKA concludes Variant of COBWEB performs better on Cloud database. Variant of COBWEB works dynamically and overcomes most of the disadvantages of k-means.

Future work of this paper includes focus on cloud service providers db and BaaS. Factor analysis is a statistical method used to describe variability among observed, correlated variables that can be used to enhance Variant of COBWEB.

References

1. Pawar A, Dani A, Enhancing Privacy-Preserving Cloud Database Querying by Preventing Brute Force Attacks, *International Journal of Computer, Information Science and Engineering*, 2014; Vol:8 No:1, p. 51-57.
2. Breetha S, Kavinila R, Hierarchical Clustering For Cancer Discovery Using Range Check And Delta Check, *International Journal of Scientific and Research Publications*, 2013; Volume 3, Issue 4.
3. Kim P, Choi J, Incremental Conceptual Clustering Using a Modified Category Utility.
4. Fisher, Douglas H, Knowledge Acquisition via incremental Conceptual Clustering, 1998; p. 139-172.
5. Halkidi M, Batistakis Y, Vazirgiannis M, On Clustering Validation Techniques, *Journal of Intelligent Information Systems*, , 2001; p.107–145.
6. Lu Y, Tsudik G, Privacy-Preserving Cloud Database Querying, Journal *of Internet Services and Information Security (JISIS)*, 2011; Vol. 1 No.4.
7. Shlens J, Tutorial on Principal Component Analysis, Google Research, 2014.
8. Wholesaler Customer Data set [online]
(https://archive.ics.uci.edu/ml/datasets/Wholesale+customers#).